J. Vis. Commun. Image R. 22 (2011) 187–200

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

A quantization-based semi-fragile watermarking scheme for image content authentication

Xiaojun Qi *, Xing Xin

Department of Computer Science, Utah State University, Logan, UT 84322-4205, United States

ARTICLE INFO

Article history:
Received 24 May 2010
Accepted 13 December 2010
Available online 16 December 2010

Keywords:
Authentication measures
Error map
Quantization
Semi-fragile watermarking
Wavelet transform
Tampered error pixels
Tampering detection sensitivity
Localization capability

ABSTRACT

This paper presents a novel semi-fragile watermarking scheme for image content authentication with tampering localization. The proposed scheme uses a non-traditional quantization method to modify one chosen approximation coefficient of each non-overlapping block to ensure its robustness against incidental attacks and fragileness against malicious attacks. The image content authentication starts with extracting watermark using the parity of quantization results from the probe image, where the round operation is used to ensure the semi-fragile property. It then constructs a binary error map and computes two authentication measures with M_1 measuring the overall similarity between extracted and embedded watermarks and M_2 measuring the overall clustering level of tampered error pixels. These two measures are further integrated to confirm the image content and localize the possible tampered areas. Our experimental results show that our scheme outperforms four peer schemes and is capable of identifying intentional tampering and incidental modification, and localizing tampered regions.

1. Introduction and related work

Trustworthy digital multimedia plays an important role in applications such as news reporting, intelligence information gathering, criminal investigation, security surveillance, and health care. However, this trustworthiness could no longer be granted since users can easily manipulate, modify, or forge digital content without causing noticeable traces using low-cost and easy-to-use digital multimedia editing software. Therefore, digital multimedia authentication has become an important issue.

Recently, digital watermarking techniques have been considered as one of the promising techniques for multimedia authentication. Among these, semi-fragile watermarking techniques have been proposed to protect copyright and prove tampering of the digital content. These techniques allow acceptable content-preserving manipulations (i.e., changing the quality of the image without modifying the image content) such as common image processing (e.g., image blurring, Gaussian low-pass filtering, median filtering, and salt and peppers noise attacks) and JPEG/JPEG2000 compression, while detecting content-altering malicious manipulations such as removal, addition, and modification of objects. We briefly review several representative semi-fragile watermarking schemes in the domain of discrete cosine transform (DCT) or discrete wavelet transform (DWT). In general, these schemes use the chosen transform domain as the media to embed and extract watermarks. They then use the extracted watermarks to authenticate the digital content and localize the tampered areas if possible.

1.1. DCT-based semi-fragile watermarking schemes

Lin et al. [1] propose to embed Gaussian distribution-based block patterns in the DCT domain. The tampering detection is made by verifying the correlation on these block patterns. This scheme can identify altered regions within a watermarked image with 75% accuracy under moderate compression and near 90% accuracy under light compression. Lin and Chang [2,3] propose to generate the invariant features at a predetermined JPEG quality factor and embed these features into mid-frequency of 8×8 DCT blocks. This scheme is robust against substitution of blocks and improves Lin et al.'s scheme [1] in that false alarms near edges hardly occur. However, they both fail to detect malicious attacks which preserve the sign of the DCT coefficients. Ho and Li [4] propose a similar yet better scheme by using the relationship of DCT coefficients in the low and middle frequencies. This scheme protects the authenticity of the compressed watermarked image when the JPEG quality is higher than their predefined lowest authenticable quality. Maeno et al. [5] propose two methods to address the shortcomings of Lin and Chang's approach [2,3]. The first method adds a random bias factor to the fixed decision boundary to catch the malicious manipulation and keep the false alarm rate low. The second method uses the non-uniform quantization scheme to

* Corresponding author. Fax: +1 435 7973265.
E-mail address: Xiaojun.Qi@usu.edu (X. Qi).

1047-3203/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2010.12.005

improve the accuracy in encoding the relationships between paired transform coefficients and increase the alteration detection sensitivity. Other representative DCT-based semi-fragile watermarking schemes include Eggers and Girod's method [6], Fridrich's method [7], and Lan et al.'s method [8].

1.2. DWT-based semi-fragile watermarking schemes

Kundur and Hatzinakos [9] embed a watermark in the quantized DWT domain. Yu et al. [10] embed a watermark in the average values of a number of wavelet coefficients in one of the detail subbands. Zhou et al. [11] propose to embed a signature from the original image into the wavelet coefficients. Kang and Park [12] incorporate the just noticeable differences feature to better discriminate malicious from non-malicious attacks. Hu and Han [13] propose to extract image features from low-frequency wavelet coefficients to generate two watermarks: one for classifying the intentional content modification and the other for indicating the modification location. Liu et al. [14] use DWT-based Zernike moments as features for the authentication task. Zhu et al. [15] apply the block-mean-based quantization strategy to embed the inter-block and intra-block signatures in the DWT domain for tamper detection and localization, respectively. Yang and Sun [16] embed the watermark by integrating the human visual system model to modify the vertical and horizontal subbands of image sub-blocks. Che et al. [17] use the dynamic quantized approach to embed watermark in low-frequency wavelet coefficients. Cruz et al. [18] employ the vector quantization method to embed a robust signature into the approximation subband of each image sub-block. However, all these schemes are only robust to moderate JPEG compression (i.e., JPEG compression of higher than 50% or 60% quality factor). The false alarm rates for watermarking schemes proposed in [9–15,18] are high under the common image processing attacks. Specifically, two schemes proposed in [9,14] achieve a 32×32 detection unit and Cruz et al.'s scheme [18] achieves a 16×16 detection unit.

In this paper, we propose a novel semi-fragile watermarking scheme by embedding a private-key-based random watermark bit sequence in the wavelet domain using the quantization method. The proposed watermarking scheme further utilizes two authentication measures derived from a binary error map to authenticate the image content and localize the tampered areas. Our proposed scheme also possesses all the desired properties for an effective authentication watermarking scheme [19], including invisibility, tamper detection, security, identification of manipulated areas, oblivion with no transmission of any secret information, and discrimination of incidental distortion and malicious tampering. Our contributions are:

- Applying the quantization method to embed the private-key-based watermark in one chosen approximation wavelet coefficient of each block so that a majority of image distortions can be detected in the authentication process.
- Defining two kinds of tampered error pixels (e.g., strongly tampered and mildly tampered error pixels) and two authentication measures to quantitatively detect the authenticity of the probe image and prove tampering.
- Using a binary error map together with the two authentication measures in the authentication process to compensate the possible misclassification in the error map, capture all possible distortions, and localize all possible tampered areas.
- Applying randomness strategies to increase the security of the proposed system.

The remainder of the paper is organized as follows: Section 2 presents the proposed semi-fragile watermarking scheme. Section 3 quantitatively evaluates the performance of the proposed scheme. Section 4 compares the proposed scheme with four peer systems [5,16–18] on extensive experiments and demonstrates the effectiveness of the proposed scheme. Section 5 draws the conclusions.

2. The proposed scheme

The proposed scheme consists of three components: watermark embedding, watermark extraction, and watermark authentication. In the following subsections, we explain each component in detail.

2.1. Watermark embedding

We divide the original image into non-overlapping 4×4 blocks and embed the private-key-based random watermark bit sequence W in the wavelet domain of each unique randomly chosen 4×4 block. The algorithmic view of the embedding procedure is shown in Fig. 1.

Here, we choose the wavelet transform domain over the other domains as the embedding media mainly due to its excellent spatial-frequency localization and its compatibility with the upcoming JPEG2000 image coding standard. We choose the low-frequency components in each 4×4 block to embed a watermark bit since high frequency components are affected by most image processing techniques and small blocks lead to high capability in localizing the possible tampered areas. Specifically, we utilize the parity of the quantized value of the approximation coefficient to embed the watermark. The parity of a value is 0 when the value is divisible by 2 and the parity of a value is 1 when the value is not divisible by 2. To ensure the watermark invisibility and increase the robustness against common image processing attacks, we choose to use one of four values (i.e., X) of the approximation subband as the media for the embedding process. The strategy of embedding a watermark bit is as follows: compute the quantized value X_q by getting the integer part of X divided by a quantizer q. If the parity of X_q equals to the embedding bit, change X to $X_q \times q$. Otherwise, change X to $X_q \times q$ plus q. All these changes ensure that the parity of the modified X is consistent with the embedding bit. It should be noted that the bigger the q, the bigger the changes, the worse the quality of the watermarked image, and the stronger the robustness. In our system, we set q as 15, which is empirically determined based on the tradeoff among invisibility, robustness, and fragileness. The randomness strategies summarized in steps 3–5 increase the security of our system. First, the order of 4×4 blocks can be easily reproduced by the same secret keys K_2 and K_3. In the meantime, the reproduction of this order is computationally infeasible without knowing K_2 and K_3. Second, the random sequence S can be easily reproduced by the same secret keys K_4, K_5, and K_6. In the meantime, the reproduction of this sequence is difficult without knowing all three secret keys. It should also be noted that the boundary check process (step 6.6) is necessary when a block is all 0's (black) or all 255's (white).

Fig. 2 illustrates the effects of embedding the watermark in the approximation subbands of the wavelet domain using the above quantization method. This figure shows that each chosen $LL_i(x, y)$'s is modified to the nearest 0 bin (the dashed line) or 1 bin (the solid line) according to its quantized value X_q and the embedding bit W_i.

2.2. Watermark extraction

The watermark extraction process uses the same one-way hash function together with the two secret keys K_2 and K_3 to choose the order of non-overlapping 4×4 blocks for extracting watermark. It then uses the parity of the quantized value X' of the approximation subband of each block to extract the watermark bit. The detailed watermark extraction steps are summarized in Fig. 3.

1. Generate a random watermark bit sequence W using the Mersenne Twister algorithm [20] and a private key K_1.
2. Divide the original image A into Len non-overlapping 4×4 blocks, when Len is the total number of blocks.
3. Apply the one-way hash function proposed in our prior robust watermarking system [21] to choose the order of the blocks for embedding watermark by using two secret keys K_2 and K_3, where K_2 is the multiplication results of two prime numbers.
4. Generate three random binary sequences of length Len using the Mersenne Twister algorithm [20] and private keys of K_4, K_5, and K_6, respectively.
5. Add these three binary sequences to get a sequence S of length Len, which contains any value of 0, 1, 2, and 3.
6. For each ordered 4×4 block B_i, its corresponding embedded watermark bit W_i, and its corresponding value S_i, perform the following operations:
   - 6.1 Apply the 1-level Haar wavelet transform to obtain the approximation subband LL_i, the horizontal subband LH_i, the vertical subband HL_i, and the diagonal subband HH_i.
   - 6.2 Choose the embedding position in $LL_i(x, y)$ using the following rules:

$$LL_i(x,y) = \begin{cases} LL_i(1,1) & \text{if } S_i = 0 \\ LL_i(1,2) & \text{if } S_i = 1 \\ LL_i(2,1) & \text{if } S_i = 2 \\ LL_i(2,2) & \text{if } S_i = 3 \end{cases} \quad (1)$$

   - 6.3 Quantize the chosen $LL_i(x, y)$ by a quantizer q using:

$$X_q = \lfloor LL_i(x,y)/q \rfloor \quad (2)$$

   - 6.4 Modify the chosen $LL_i(x, y)$ value by:

$$LL_i(x,y) = \begin{cases} X_q \times q & if\ \mathrm{mod}(X_q,2) == W_i \\ X_q \times q + q & otherwise \end{cases} \quad (3)$$

   where mod(X_q, 2) computes the remainder of X_q divided by 2.
   - 6.5 Apply the inverse 1-level Haar wavelet transform to obtain the watermarked block.
   - 6.6 Perform the boundary check on the 2×2 corner of the watermarked block to ensure its four values are in the proper range. This 2×2 corner corresponds to the upper-left, upper-right, lower-left, and lower-right corner of the watermarked block when the value of S_i is 0, 1, 2, and 3, respectively. For an 8-bit grayscale image, the proper range is [0- q/4, 255+q/4]. If any of the four values in the 2×2 corner falls outside of the proper range, apply the following remedy strategies:
     a) If one value is larger than the upper-bound of the allowable range, modify chosen $LL_i(x, y)$ by:

$$LL_i(x,y) = X_q \times q - q \quad (4)$$

     b) If one value is smaller than the lower-bound of the allowable range, modify chosen $LL_i(x, y)$ by:

$$LL_i(x,y) = X_q \times q + 2 \times q \quad (5)$$

     c) Apply the inverse 1-level Haar wavelet transform to obtain the corrected watermarked block. If any value in the 2×2 corner of the corrected watermark block falls outside of the proper range, modify its value by adding or subtracting 4×q to ensure the modified value is in the proper range and the parity of X_q is intact.

Fig. 1. The algorithmic view of the watermark embedding procedure. (See above-mentioned references for further information.)

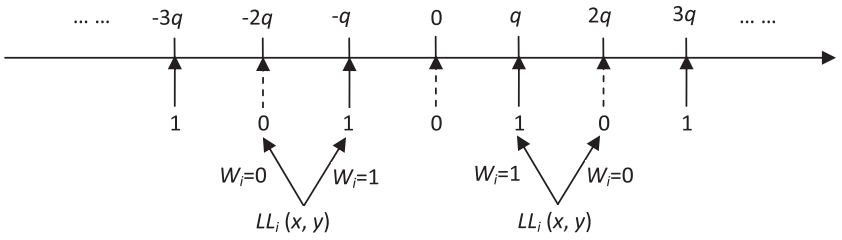

Fig. 2. Illustration of the quantization process.

1. Divide the probe image A' into non-overlapping 4×4 blocks.
2. Generate a sequence of S using the same strategy as used in the embedding process together with the same secret keys of K_4, K_5, and K_6.
3. For each 4×4 block B_i', whose order is chosen by using the same one-way hash function as used in the embedding process together with the same secret keys of K_2 and K_3, perform the following operations:
   - 3.1 Apply the 1-level Haar wavelet transform to obtain the approximation subband LL_i', the horizontal subband LH_i', the vertical subband HL_i', and the diagonal subband HH_i'.
   - 3.2 Choose the extracting position in LL_i'(x, y) using the same rules specified in (1).
   - 3.3 Quantize the chosen LL_i'(x, y) by the same quantizer q using:

$$X_q' = \mathrm{round}(LL_i'(x,y)/q) \quad (6)$$

   - 3.4 Set the extracted watermarked bit EW_i as mod(X_q', 2).

Fig. 3. The algorithmic view of the watermark extraction procedure.

2.3. Watermark authentication

We generate a binary error map to perform the watermark authentication task. Since the extracted watermark *EW* reflects the changes of local intensity resulting from attacks, we construct the error map, i.e., *ErrorMap*, by mapping the absolute difference between EW_i and W_i (e.g., $|EW_i - W_i|$) onto its corresponding 4×4 block. The 0's and 1's in *ErrorMap* indicate the match and mismatch between extracted and embedded watermarks, respectively. In other words, any pixel with the value of 1's in *ErrorMap* is an error pixel. In our system, we classify the error pixels into three categories: strongly tampered, mildly tampered, and isolated error pixels. Fig. 4 illustrates these three categories of error pixels in red solid circles using a window size of 3×3. Specifically, we consider an error pixel as strongly tampered if at least four of its eight neighbors are error pixels (marked by black solid circles); an error pixel as mildly tampered if one, two, or three of its eight neighbors are error pixels; and an error pixel as isolated (e.g., likely caused by noise) if none of its eight neighbors is an error pixel. Since any malicious attack normally tries to modify the image content (e.g., smoothly remove/modify an object in an image and/or add an object in an image) without causing any suspicion from the image owner and sparsely manipulating individual pixels in an image will not modify the image content, we do not consider the isolated error pixel as the tampered error pixel and consider both strongly tampered and mildly tampered error pixels as tampered error pixels. It should be noted that the window size and thresholds of the number of neighboring error pixels in the window for defining strongly tampered and mildly tampered error pixels can be set differently based on the specific application requirement. They also determine the sensitivity of the authentication process.

We then define two authentication measures, M_1 and M_2, to protect copyright and prove tampering, where M_1 measures the overall similarity between extracted and embedded watermarks and M_2 measures the overall clustering level of the tampered error pixels. We compute M_1 as the percentage of error pixels (i.e., 1's) in *ErrorMap*. We compute M_2 as the ratio of the number of strongly tampered error pixels to the number of tampered error pixels in *ErrorMap*. The value of M_2 is set to 0's when the number of tampered error pixels is zero.

Finally, we design a quantitative method to decide the authenticity of the probe image based on our defined two authentication measures. The algorithmic view of the authentication process is summarized in Fig. 5.

It is important to apply the median filtering on *ErrorMap* when M_1 is less than or equal to 0.1 (i.e., at most 10% of 4×4 blocks are detected as distorted). Due to the small amount of distortions, we can infer that the probe image must have undergone small malicious attacks or moderate incidental attacks. That is, the tampered regions would be small and tend to cluster if malicious attacks occurred and the tampered regions would be small and tend to scatter if moderate incidental attacks occurred. This median filtering removes all mildly tampered error pixels. It also turns non-error pixels as error pixels if the non-error pixels are surrounded by at least five error pixels. That is, this filtering keeps the clustered error pixels intact and makes scattered mildly tampered error pixels and isolated error pixels disappear. As a result, the small malicious attack leads to a larger M_2 value due to the removal of mildly distorted error pixels. Our extensive experiments show that the value of 0.1 for T_{median} works well on all 200 test images and all the simulated attacks.

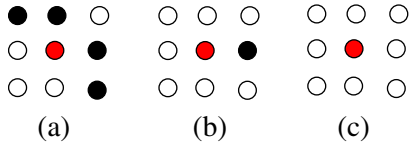

Fig. 4. Examples of three categories of error pixels shown in red solid circles. (a) Strongly tampered error pixel. (b) Mildly tampered error pixel. (c) Isolated error pixel. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

1. Compute M_1 using *ErrorMap*
2. if $M_1 \le T_{median}$ (i.e., 0.1)
   Update *ErrorMap* as its 3×3 median filtering result
   endif
3. Compute M_2 using *ErrorMap*
4. if $0 \le M_1 < T_{halferrorbit}$
   if $M_2 < T_{malicious}$, the probe image is authenticated
   else the probe image is maliciously attacked
5. if $T_{halferrorbit} \le M_1 < T_{errorbit}$
   if $M_2 < T_{malicious}$, the probe image is incidentally attacked
   else the probe image is maliciously attacked
6. if $M_1 \ge T_{errorbit}$, the probe image is not embedded with watermarks

Fig. 5. The algorithmic view of the authentication process.

The remaining thresholds, i.e., $T_{errorbit}$, $T_{halferrorbit}$, and $T_{malicious}$, involved in the authentication process are determined based on the predefined false negative probability of 10^{-6}. The threshold for $T_{errorbit}$ is derived as follows: A pixel in *ErrorMap* can have a value either 0's or 1's. When the embedded watermark is a private-key-based random watermark bit sequence and the original watermark bit sequence is not embedded in the image in question, the probability that the extracted watermark bit sequence will match the original watermark bit sequence can be considered to be 0.5 for each bit. This is equivalent to the tossed coin problem, which inherently has only two possible and equally likely outcomes. As a result, the probability for a pixel in *Errormap* to be 0 or 1 is 0.5 and each pixel is a binomially distributed random variable. The expected value (i.e., $E(errorbit)$) and the variance of error bits (i.e., $Var(errorbit)$) are respectively $0.5 \times numel$ and $0.25 \times numel$, where *numel* is the total number of pixels in *ErrorMap*. Therefore, we deduce the threshold for detecting if the probe image has been embedded with the watermark bits by:

$$\begin{aligned} P(T_{errorbit} \geqslant \tau_1) &= 1 - P(T_{errorbit} < \tau_1) \\ &\approx 1 - \Phi\left[\frac{\tau - E(errorbit)}{\sqrt{Var(errorbit)}}\right] \geqslant 1 - 10^{-6} \\ &\Rightarrow \tau_1 \geqslant 7924.9 \Rightarrow T_{errorbit} \geqslant \frac{7924.9}{numel} = 0.4837 \end{aligned} \tag{7}$$

Here, Φ approaches the normal distribution with expected value 0 and variance 1 when *numel* is large (refer to Appendix A for proof). Hence, we consider that the image is not embedded with our watermark if $M_1 \geqslant T_{errorbit} = 0.4837$ and use a half of $T_{errorbit}$ as $T_{halferrorbit}$, which is the threshold for distinguishing incidentally attacked watermarked images from authenticated watermarked images.

The threshold for $T_{malicious}$ is derived as follows: The probability for a pixel to be detected as tampered error pixels is $\frac{1}{2}[1 - (\frac{1}{2})^8] = 255/512 = 0.4980$. The probability for a pixel to be detected as strongly tampered error pixel is $\left(C_8^4 + C_8^5 + C_8^6 + C_8^7 + C_8^8\right) \times 0.5^9 = 163/512 = 0.3184$. Then the expected value (i.e., $E(strongtampix)$) and the variance of strongly tampered error pixels (i.e., $Var(strongtampix)$) are $163/512 \times numel$ and $163/512 \times (1 - 163/512) \times numel = 0.2170 \times numel$, respectively. The expected value of tampered error pixels (i.e., $E(tampix)$) is $0.4980 \times numel$. Therefore, we deduce the threshold for detecting malicious attacks by:

$$P(T_{malicious} \geqslant \tau_2) = 1 - P(T_{malicious} < \tau_2)$$
$$\approx 1 - \Phi\left[\frac{\tau - E(strongtampix)}{\sqrt{Var(strongtampix)}}\right] \geqslant 1 - 10^{-6}$$
$$\Rightarrow \tau_2 \geqslant 4964.9 \Rightarrow T_{malicious} \geqslant \frac{4964.9}{E(tampix)} = 0.6085 \quad (8)$$

Here, Φ approaches the normal distribution with expected value 0 and variance 1 when *numel* is large (refer to Appendix A for proof). Hence, we consider the attack on the watermarked image is malicious if $M_2 \geqslant T_{malicious} = 0.6085$.

2.4. Validation of the defined error pixels and authentication measures

Our definitions of the three categories of error pixels and the two authentication measures are guided by the following observations. (1) Most error pixels would spread across the error map if incidental attacks were made on the watermarked image. (2) Most error pixels would cluster in distorted regions if malicious attacks were made on the watermarked image. Fig. 6 demonstrates these two observations by showing the error pixel distribution after performing two attacks (e.g., obvious malicious attack by adding a black square and JPEG compression attack with the 70% quality factor) on the standard watermarked "Lena" image, respectively. For the error pixel distribution under each attack, we sequentially display the distribution of all error pixels (i.e., *ErrorMap*), tampered error pixels, and strongly tampered error pixels. We clearly observe the following: (1) Fig. 6(a) shows that *ErrorMap* contains exclusively clustered tampered error pixels when the malicious attack is applied to the watermarked image. The strongly tampered error pixels are also clustered within the tampered areas under this malicious attack. (2) Fig. 6(b) shows that *ErrorMap* contains a majority of randomly spread tampered error pixels when the incidental attack is applied to the watermarked image. The strongly tampered error pixels tend to be isolated under the incidental attack. Based on the predefined thresholds, our system successfully detects each watermarked image shown in Fig. 6(a) and (b) as maliciously attacked and incidentally attacked, respectively.

3. Performance analysis

In the following, we quantitatively evaluate the performance of the proposed scheme in terms of the quality of the watermarked image, the effects of the quantizer q, the tampering detection sensitivity, and the localization capability.

3.1. The quality of the watermarked image

In the proposed scheme, the image distortion is caused by the modifications of the wavelet coefficients in the embedding process. Both the quantizer q and the watermark payload p (i.e., the number of watermark bits embedded in the host image) affect the quality of the watermarked image. In the following, we derive the mean squared error (MSE) incurred in the embedding process. Since there is roughly an equal distribution of all values in the approximation subband (i.e., the statistics for the approximation subband differ significantly from the detail subbands) [22], we assume that the original wavelet coefficients in the approximation subband are uniformly distributed over the range of $[kq, (k+1)q]$ for $k \in Z$. When the parity of the quantization result of the original wavelet coefficient $LL_i(x, y)$ matches the embedded watermark bit W_i, $LL_i(x, y)$ is modified to the lower-bound kq, and the MSE caused by this quantization is:

$$\mathrm{MSE}_1 = \frac{1}{q}\int_0^q \tau^2 d\tau = \frac{q^2}{3} \quad (9)$$

Otherwise, $LL_i(x, y)$ is modified to the upper-bound $(k+1)q$ and the MSE caused by this quantization is:

$$\mathrm{MSE}_2 = \frac{1}{q}\int_0^q (\tau - q)^2 d\tau = \frac{q^2}{3} \quad (10)$$

As a result, the average distortion caused by embedding one watermark bit is $q^2/3$ and the MSE of embedding p watermark bits in the block-based wavelet domain is:

$$\mathrm{MSE} = \frac{p \times q^2}{3 \times W \times H} \quad (11)$$

where W and H are the width and the height of the host image, respectively. According to the Parseval's theorem, the MSE of the entire image equals to its counterpart in the wavelet domain. Therefore, the *PSNR* value of the watermarked image is:

$$PSNR = 20\log_{10}\left(\frac{255}{q}\sqrt{\frac{3 \times W \times H}{p}}\right)$$
$$= 20\log_{10}\left(\frac{255}{q}\sqrt{\frac{3 \times W \times H}{\frac{W \times H}{bs \times bs}}}\right)$$
$$= 20\log_{10}\left(\frac{255 \times bs}{q}\sqrt{3}\right) \quad (12)$$

where bs denotes the size of each embedding square block. It clearly reveals that the quality of the watermarked image is determined by p and q. Smaller p's and q's lead to larger *PSNR* values. In our system, we set q as 15 and chose block size bs as 4. Therefore, p equals to $W \times H/16$ and the expected *PSNR* value is around 41.42 db. Based on (12), the expected *PSNR* values are 44.12, 42.66, 40.33, and 39.37 for quantizers of 11, 13, 17, and 19, respectively.

Our experimental results on 200 8-bit grayscale images show that the average *PSNR* values of their watermarked images by using our scheme with quantizers of 11, 13, 15, 17, and 19 are 43.86, 42.11, 40.68, 39.97, and 38.99, respectively. These averages are consistent with our computed expected values and are higher than the empirical value (e.g., 35.00 db) for the image without perceivable degradation [23]. These averages also confirm that the quality of the watermarked image degrades as the quantizer increases and our choice of the quantizer of 15 achieves excellent quality with a high *PSNR* value of above 40 db.

(a) $M_1 = 0.0325$ and $M_2 = 0.8218$ (b) $M_1 = 0.0542$ and $M_2 = 0$

Fig. 6. Illustration of the error pixel distribution. (a) The maliciously attacked watermarked image. (b) The 70% JPEG compressed watermarked image. Along with their corresponding *ErrorMap's* in terms of all error pixels, tampered error pixels, and strongly tampered error pixels. The size of the error map is enlarged for easy reading of error pixels.

3.2. The effects of the quantizer q

To further validate the choice of the quantizer, we tested the performance of the proposed scheme under various JPEG lossy compression attacks using different quantizers (e.g., 11, 13, 15, 17, and 19). Fig. 7 shows the average values of M_1's and M_2's of 200 watermarked images after no attack and after ten levels of JPEG compression with a quality factor of 100% down to 10% with a step size of 10%. We also show the two threshold lines of 0.4837 and 0.2418 for M_1 and the threshold line of 0.6085 for M_2 to facilitate comparison. The figure clearly demonstrates the following: (1) The M_1 and M_2 values are 0's under no attack and after the JPEG compression of 100% quality factor. (2) For each chosen quantizer, the M_1 and M_2 values increase as the JPEG quality factor decreases. (3) For each JPEG compression quality factor, the M_1 and M_2 values increase as the quantizer decreases. (4) For all quantizers, the M_1 value is below the threshold of 0.2418 for quality factors higher than 60% and the M_2 value is below the threshold of 0.6085 for quality factors higher than 10%. To ensure our scheme achieves excellent watermark invisibility with the *PSNR* value around 41 db and is robust to JPEG compression of quality factors higher than 50%, we choose q as 15.

3.3. The tampering detection sensitivity

The tampering detection sensitivity of the proposed scheme is determined by the quantizer. The error map captures the changes in the quantization results and makes the tampering detectable for $k \in Z$ in the following two cases:

(1) The wavelet coefficient $LL'_i(x,y)$ of the watermarked image is $2kq$ and the manipulation causes a shift of $LL'_i(x,y)$ in the range of $[(0.5+2k)q, (1.5+2k)q)$.
(2) The wavelet coefficient $LL'_i(x,y)$ of the watermarked image is $2kq+q$ and the manipulation causes a shift of $LL'_i(x,y)$ in the range of $[(1.5+2k)q, (2.5+2k)q)$.

That is, our scheme is capable of detecting all the changes satisfying the above two conditions. Small changes of a half of the quantizer q or other changes falling in the range of $[(-0.5+2k)q, (0.5+2k)q]$ in the distorted area do not modify the parity of the quantized approximation value. As a result, our scheme is robust to moderate image content preserving attacks which do not dramatically change the pixel intensity. However, some pixels in the tampered area may be missed when the changes in the wavelet domain do not satisfy the above two conditions. To address this shortcoming, our authentication process utilizes the distribution of the detected error pixels to evaluate the authenticity of the probe image. Specifically, the observations shown in Fig. 6 are incorporated to compensate the possible misclassification in *ErrorMap*.

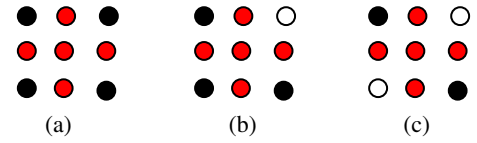

Fig. 8. Illustration of different error pixels distributions in 3×3 blocks. (a) Nine error pixels with 5 strongly tampered error pixels (1 case and $M_2 = 0.56$). (b) Eight error pixels with 5 strongly tampered error pixels (4 cases and $M_2 = 0.625$). (c) Seven error pixels with 5 strongly tampered error pixels (2 cases and $M_2 = 0.714$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The tampering detection sensitivity can also be adjusted by choosing different window sizes and thresholds of the number of neighboring error pixels in the window for defining strongly and mildly tampered error pixels. If the threshold is preset, the larger the window size, the lower the sensitivity. Based on the application requirements, the proposed scheme can identify various tampered areas and detect bigger alterations whereas bypassing smaller alterations using a predetermined window size.

3.4. The localization capability

In the proposed scheme, the image content is monitored by the embedded and extracted watermark. Specifically, changing a value in the 2×2 corner of each 4×4 block may result in a mismatch in *ErrorMap*. To compensate the misclassification, we employ a window size of 3×3 to categorize each non-isolated error pixel as either strongly tampered or mildly tampered, as defined in Section 2.3. The localization capability refers to the capability to find the smallest tampered area (a.k.a, the detection unit) in a probe image. Here, we start our analysis with any possibly smallest 3×3 block in *ErrorMap*, where all the nine pixels in the block are error pixels and all the pixels outside of the block are non-error pixels (shown in Fig. 8(a)). Based on our definition of three kinds of error pixels, we know that the five error pixels marked by solid red circles are strongly tampered error pixels and all the nine error pixels in the 3×3 block are tampered error pixels. Therefore, $M_2 = 5/9 = 0.5556$, which is less than our threshold of $T_{malicious}$ (i.e., 0.6085), and we conclude that the 3×3 block is not a maliciously tampered area. However, the values of M_2 are 0.625 and 0.714 when the error pixel distributions follow the sample patterns shown in Fig. 8(b) and (c), respectively. That is, our scheme can achieve a 12×12 detection unit when the error pixels follow the sample distributions shown in Fig. 8(b) and (c).

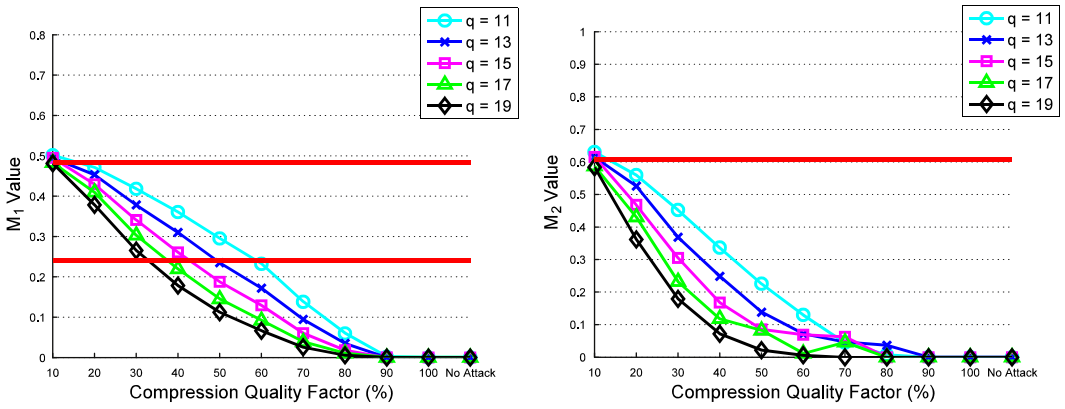

Fig. 7. Illustration of the effects of five quantizers on the M_1 and M_2 values.

4. Experimental results

To evaluate the performance of the proposed semi-fragile watermarking scheme, we first compared the quality of the watermarked images of our scheme and four peer schemes, namely, Maeno et al.'s scheme using the random bias [5], Yang and Sun's scheme [16], Che et al.'s scheme [17], and Cruz et al.'s scheme [18], using five representative 8-bit 512×512 grayscale images and 200 8-bit grayscale images including 30 commonly used 8-bit grayscale images and 170 8-bit grayscale images converted from our personally collected pictures. We then conducted extensive experiments on 200 8-bit grayscale images by comparing our system with these four peer systems using their suggested empirically determined parameters. Different kinds of attempting manipulations were simulated. These simulated manipulations include the following: ten levels of image blurring, ten levels of Gaussian low-pass filtering, ten levels of median filtering, five levels of salt and peppers noise addition, ten levels of JPEG lossy compression, ten levels of JPEG2000 lossy compression, three kinds of Photoshop modifications (e.g., add an object, delete an object, and modify an object), and three levels of substitution attacks followed by JPEG compression of the 80% quality factor. Specifically, we consider the substitution attacks [24], which substitute the semi-fragile watermarked image with its original un-watermarked version, as representative of malicious attacks since the variety of forgery attacks can be collapsed to substitution attacks. Furthermore, substitution of an image block with the original version is the most difficult to detect forgery attack. To this end, we randomly chose a 24×24 block or a 36×36 block or a 48×48 block from the original image to replace a random block in its watermarked image. Finally, we evaluated the performance of our proposed scheme, the variant of our proposed scheme in the spatial domain, and four peer systems under two *PSNR* values (e.g., 38 and 41) for all simulated attacks and each specific category of simulated attacks, respectively. For the substitution attacks, we performed each of the three kinds of block substitutions followed by 80% JPEG compression 50 times on each of the 200 watermarked images.

To ensure fair comparison, we carefully studied the authentication process of each scheme to find its equivalent measure(s) as those used in our scheme. The values of these authentication measures can be further used to compare the sensitivity of the semi-fragile watermarking schemes under a variety of common image processing attacks and malicious attacks. That is, the best semi-fragile watermarking scheme should lead to the smallest values and the smallest changes in authentication measures under incidental attacks and should lead to relatively large values in the M_2 equivalent authentication measure under malicious attacks. We found that all four peer schemes used a measure similar to our M_1's in their authentication process. Yang's scheme also used another measure similar to our M_2's. In addition, Yang's scheme explicitly summarized the thresholds for detecting a probe image as authentic, incidentally distorted, or maliciously distorted. The other three schemes did not explicitly mention the thresholds for their decision making. However, we can roughly infer their thresholds from their discussions. These thresholds are around 0.3 and are a little bit higher than $T_{halferrorbit}$ (i.e., 0.2418) for M_1's in our scheme. All four peer schemes visually showed the error maps or the localization results without listing the values of their authentication measures. That is, they all reply on the visual inspection to show the effectiveness of their localization results. In our experiments on various malicious attacks, we show both the values of the authentication measures and the localization results to validate the effectiveness of our scheme.

4.1. Watermark invisibility

Table 1 summarizes the *PSNR* values after embedding watermarks in five representative images and the average *PSNR* values after embedding watermarks in 200 test images using our scheme and four peer schemes, respectively. This table clearly shows that our *PSNR* values for five representative images and our average *PSNR* values for 200 test images are larger than 40.00 db and are comparable with the expected *PSNR* value computed in Section 3.1. Overall, our *PSNR* values are also higher than or comparable to the *PSNR* values of four peer schemes expect Cruz et al.'s scheme [18], which embeds watermark bits in larger blocks of 16×16.

4.2. Robustness to common image processing attacks

We performed four kinds of representative image processing attacks on 200 watermarked images. These attacks are ten levels of image blurring attacks using circular averaging filters of radii of 1.1–2 with an increasing step size of 0.1, ten levels of Gaussian low-pass filtering attacks using rotationally symmetric Gaussian low-pass filters of size 3×3 and standard deviation ranging from 0.1 to 1 with an increasing step size of 0.1, ten levels of median filtering attacks using filters of radii of 3–12 with an increasing step size of 1, and five levels of salt and peppers noise attacks using noise density ranging from 0.01 to 0.05 with an increasing step size of 0.01. Since all four peer schemes used a measure similar to our M_1's in the authentication process, we plot the average M_1 values of 200 watermarked images under each image processing attack for all five schemes on the left side of Fig. 9. Yang's scheme also used another measure similar to our M_2's in the authentication process. As a result, we plot the average M_2 values of 200 watermarked images under each image processing attack for these two schemes on the right side of Fig. 9.

Fig. 9 clearly shows the following: (1) all the average values of our M_2's are below the threshold line of 0.6085 and are significantly smaller than Yang's M_2's for all incidental attacks; (2) all the average values of our M_1's are the smallest under the same incidental attack; (3) all the average values of our M_1's have the smallest changes when incidental attacks decrease the image quality without altering the image content. This indicates that our scheme is more robust in classifying a watermarked image under any of these image processing attacks as authentic or incidentally distorted. Specifically, our scheme successfully detects all 200 watermarked images under Gaussian low-pass filtering, salt and peppers noise addition, or image blurring attacks with circular averaging filters of radii smaller than or equal to 1.4 as authentic. The watermarked image under blurring, Gaussian low-pass filtering, or salt

Table 1
Comparison of *PSNR* values.

	Lena	Peppers	Baboon	Airplane	Cameraman	Average of 200 images
Ours	41.04	40.51	41.30	40.35	40.18	40.54
Yang	38.28	37.05	32.76	35.70	42.59	36.72
Che	39.45	37.95	37.84	37.64	38.76	37.43
Maeno	32.42	31.56	31.29	31.02	32.64	31.65
Cruz	45.78	45.29	45.09	44.51	46.10	44.32

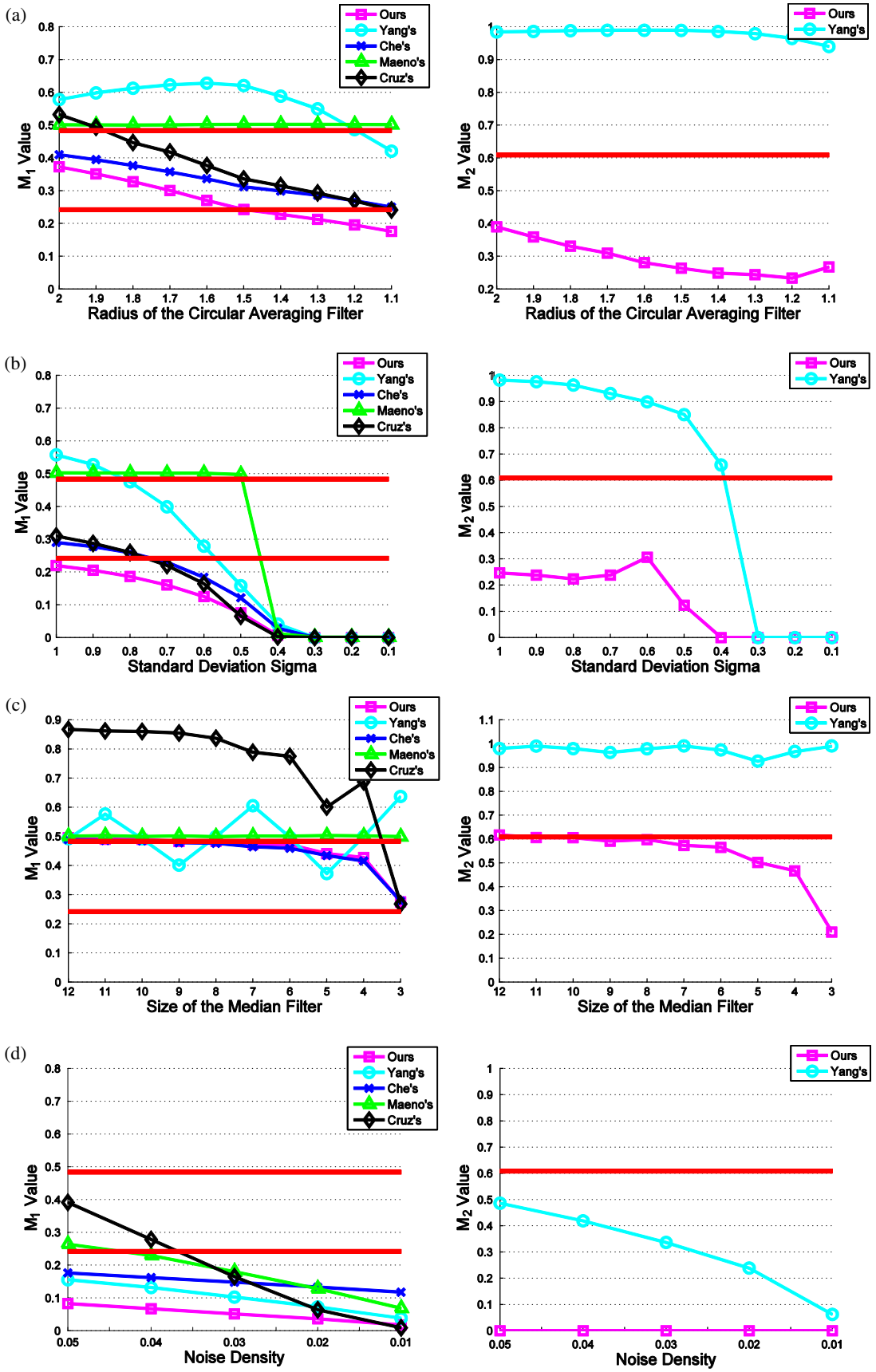

Fig. 9. Comparison of various common image processing attacks on M_1's (left) of our scheme and four peer schemes and M_2's (right) of our scheme and Yang's scheme. (a) Image blurring attacks. (b) Gaussian low-pass filtering attacks. (c) Median filtering attacks. (d) Salt and peppers noise attacks.

and peppers noise attacks is detected as authenticated if its M_1 value is smaller than $T_{halferrorbit}$ (0.2418) and as incidentally distorted if its M_1 value is between $T_{halferrorbit}$ (0.2418) and $T_{errorbit}$ (0.4837). The watermarked image under median filtering attacks with a filter size ranging from 3 to 9 is detected as incidentally distorted since its M_1 value is between $T_{halferrorbit}$ and $T_{errorbit}$. However, our scheme detects the watermarked image under median filtering attacks with a larger filter size as a non-copyrighted image since its M_1 value is larger than 0.4837. This is reasonable due to its significant changes on the watermarked image.

4.3. Robustness to JPEG lossy compression and JPEG2000 lossy compression attacks

We performed two kinds of compressions, namely, conventional JPEG lossy compression and JPEG2000 lossy compression, on 200 watermarked images. The left plot of Fig. 10 compares the average values of M_1's of 200 watermarked images under no attack and various JPEG compression attacks using quality factors of 100% down to 10% with a decreasing step size of 10% of all five schemes. The right plot of Fig. 10 compares the average values of M_2's of 200 watermarked images under no attack and the same ten levels of JPEG compression attacks of our scheme and Yang's scheme. Fig. 10 clearly shows that all the average values of our M_2's are much smaller than the corresponding average values of Yang's scheme and are below the threshold line of 0.6085 for JPEG compressions of a quality factor down to 10%. That is, the watermarked image under JPEG compressions of a quality factor down to 10% is detected as authentic if its M_1 value is smaller than 0.2418 and as incidentally distorted if its M_1 value is between 0.2418 and 0.4837. In addition, our average M_1 values generally are much smaller than the corresponding average values of four peer schemes for JPEG quality factors down to 30%. Our scheme is also the only one that achieves lower than the threshold value of 0.2418 for M_1's for JPEG quality factors down to 50%. All these indicate that our scheme is more robust in classifying a watermarked image under JPEG compressions of at least 50% quality factor as authentic and classifying a watermarked image under JPEG compressions of a quality factor ranging from 10% to 50% as incidentally distorted. Our experimental results on 200 watermarked images also confirm the above indication. None of the four peer schemes achieves the comparable performance as our scheme. Specifically, they detect the watermarked images under JPEG compressions of at least 60% quality factor as incidentally distorted or authentic and detect the watermarked images under JPEG compressions of 10–50% quality factors as maliciously distorted.

To evaluate the robustness of our proposed scheme to JPEG2000 lossy compression attacks, we further compare the average values of M_1's and M_2's of 200 watermarked images under various JPEG2000 compression attacks using quality factors of 1000% down to 100% with a decreasing step size of 100% and their equivalent JPEG compression attacks using quality factors of 100% down to 10% with a decreasing step size of 10% in Fig. 11. We also plot the values of M_1's and M_2's under each attack after adding or subtracting the standard deviation values (STDV) from their average values. It clearly shows that all the average values of M_1's and M_2's for JPEG2000 compression attacks are much smaller than the ones for JPEG compression attacks. The relationship holds true for the average values of M_1's and M_2's adding or subtracting their corresponding STDVs. In addition, the values of M_2's are below the

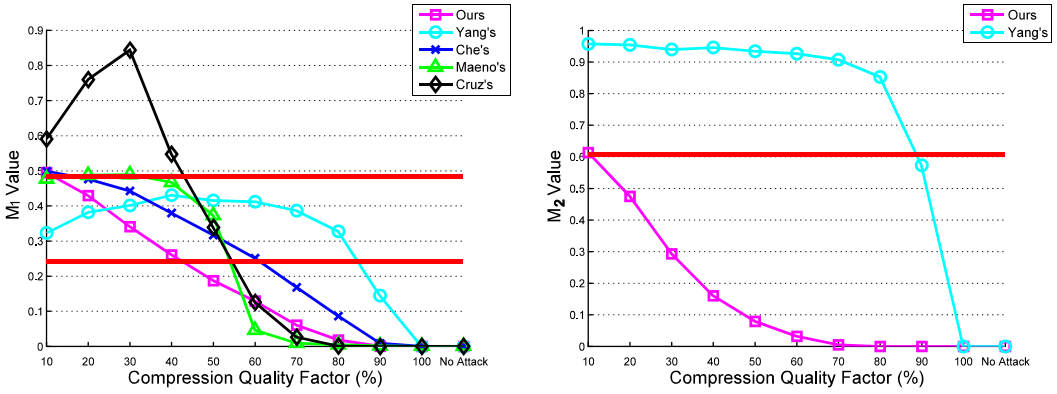

Fig. 10. Comparison of various JEPG compression attacks on M_1's (left) of our scheme and four peer schemes and M_2's (right) of our scheme and Yang's scheme.

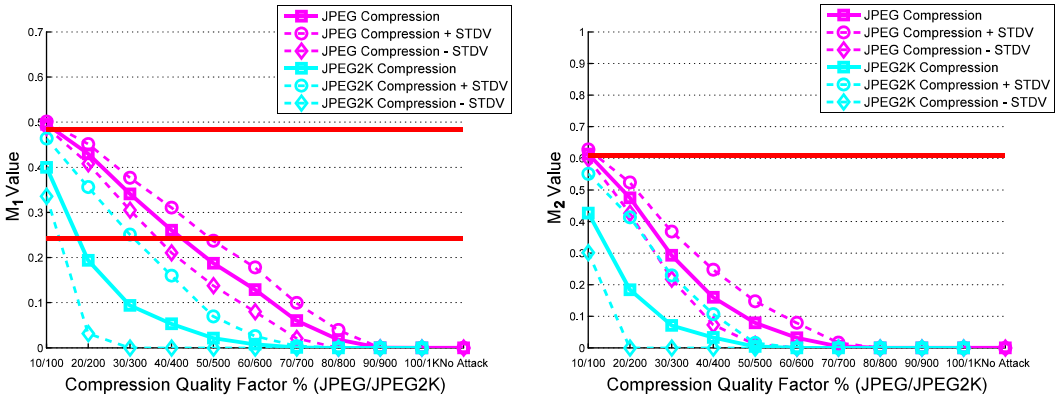

Fig. 11. Comparison of various JEPG2000 compression attacks and their corresponding JPEG compression attacks on M_1's (left) and M_2's (right) of our scheme.

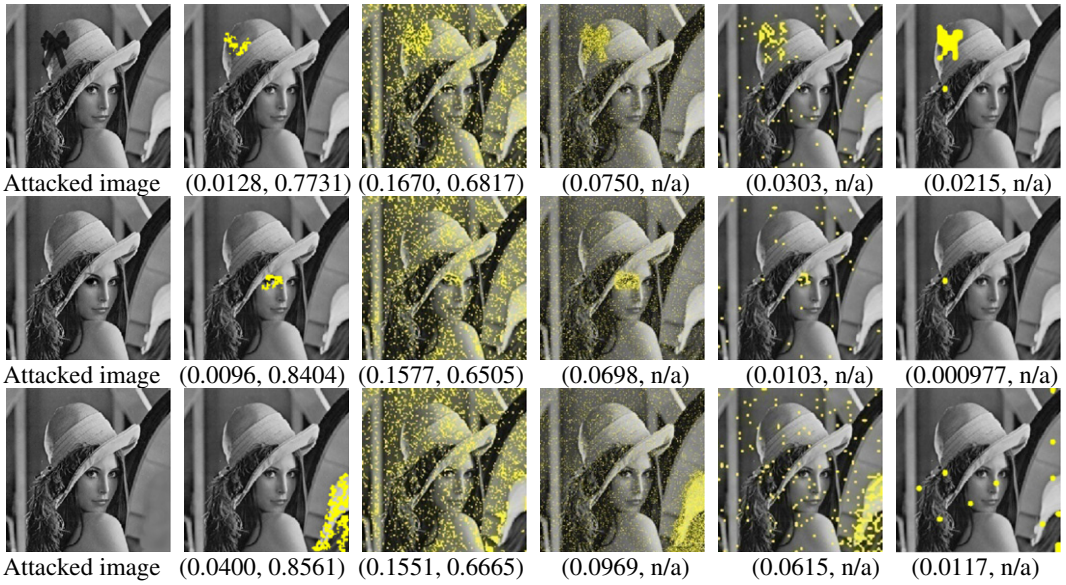

Fig. 12. Comparison of our, Yang's, Che's, Maeno's, and Cruz's (from left to right) localization results after realistic malicious attacks. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

threshold line of 0.6085 for all JPEG2000 compressions and the values of M_1's are below the threshold line of 0.2418 for all JPEG2000 compressions except the ones with the quality factor of 100%, 200%, and 300%. That is, the watermarked image under JPEG2000 compressions of a quality factor down to 400% is detected as authentic. Our experimental results also clearly demonstrate that

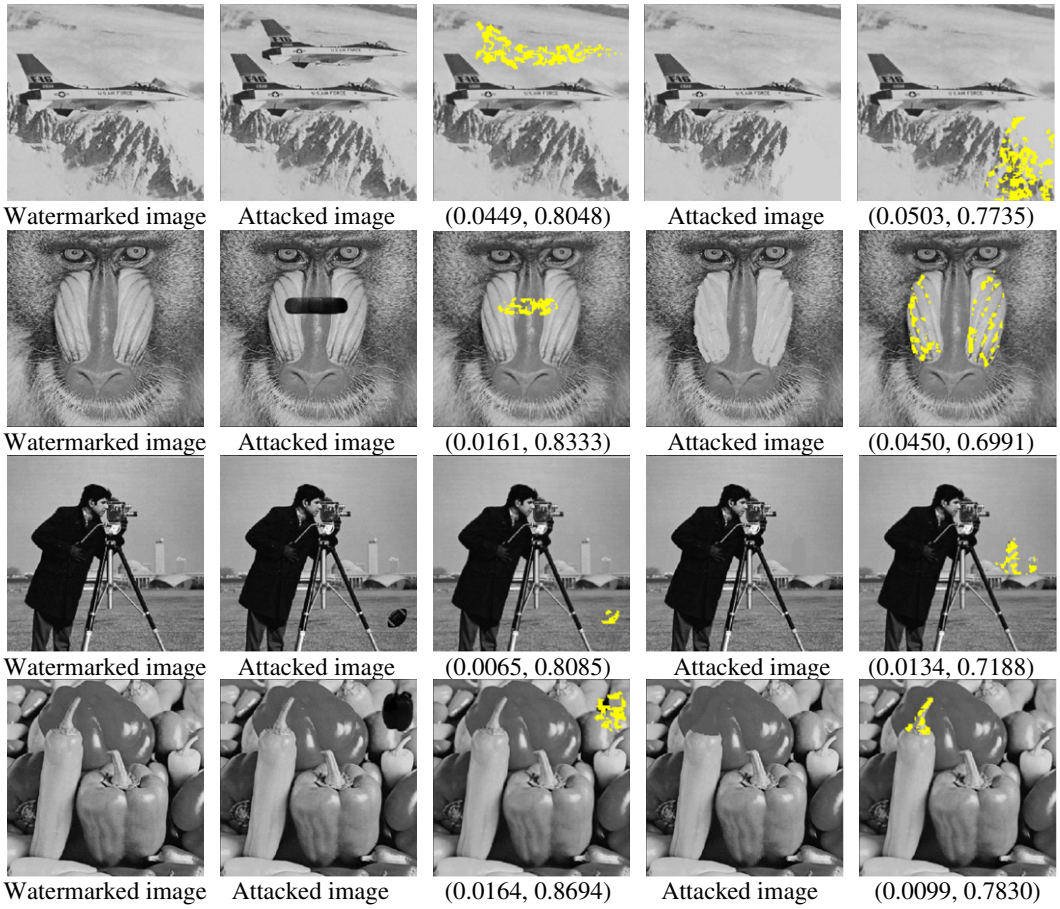

Fig. 13. Illustration of our tamper localization results: watermarked images (1st column), maliciously attacked images by inserting an external object (2nd column), our corresponding detected distortion regions (3rd column), maliciously attacked images by modifying or removing an object (4th column), and our corresponding detected distortion regions (5th column). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

our proposed scheme is more robust against JPEG2000 compression attacks than JPEG compression attacks since it works in the wavelet domain, which is the same domain that JPEG2000 compression works in.

4.4. Fragileness to various malicious attacks

We performed various malicious attacks on 200 watermarked images to demonstrate the effectiveness of our proposed scheme in localizing the maliciously tampered regions. To this end, we applied three kinds of realistic modifications on the watermarked "Lena" image by using Photoshop to insert an external object (decoration on hat), modify the right eye, and remove the object (white and gray wavy decoration) on the lower right, respectively. The maliciously attacked "Lena" image was then saved as a JPG image using the default compression setting. Fig. 12 demonstrates the localization results, shown in yellow, of five schemes and lists the M_1 and M_2 values, whenever applicable, in a pair for each scheme. The figure clearly shows that our scheme achieves the best and the cleanest localization results and Maeno's scheme achieves the second best localization results with a few additional small isolated distorted regions resulting from the JPEG compression. Che's scheme achieves comparable localization results as Maeno's scheme except that it detects more distorted regions resulting from the JPEG compression due to its less robustness to JPEG compression. Based on our thresholds for two authentication measures, we conclude that our scheme detects all three maliciously attacked watermarked "Lena" image as maliciously tampered and correctly localizes their tampered regions. Yang's scheme detects these maliciously attacked watermarked images as maliciously distorted. However, it does not produce a decent localization result under any of the three malicious attacks due to its less robustness to JPEG compressions. The other three schemes obtain small values for M_1's, which are similar to the values obtained under image processing and JPEG compression attacks. As a result, they detect these maliciously attacked images as authentic based on our equivalent predefined thresholds.

Fig. 13 demonstrates the tampering localization results on four additional representative watermarked images, which were maliciously attacked by inserting an external object or removing

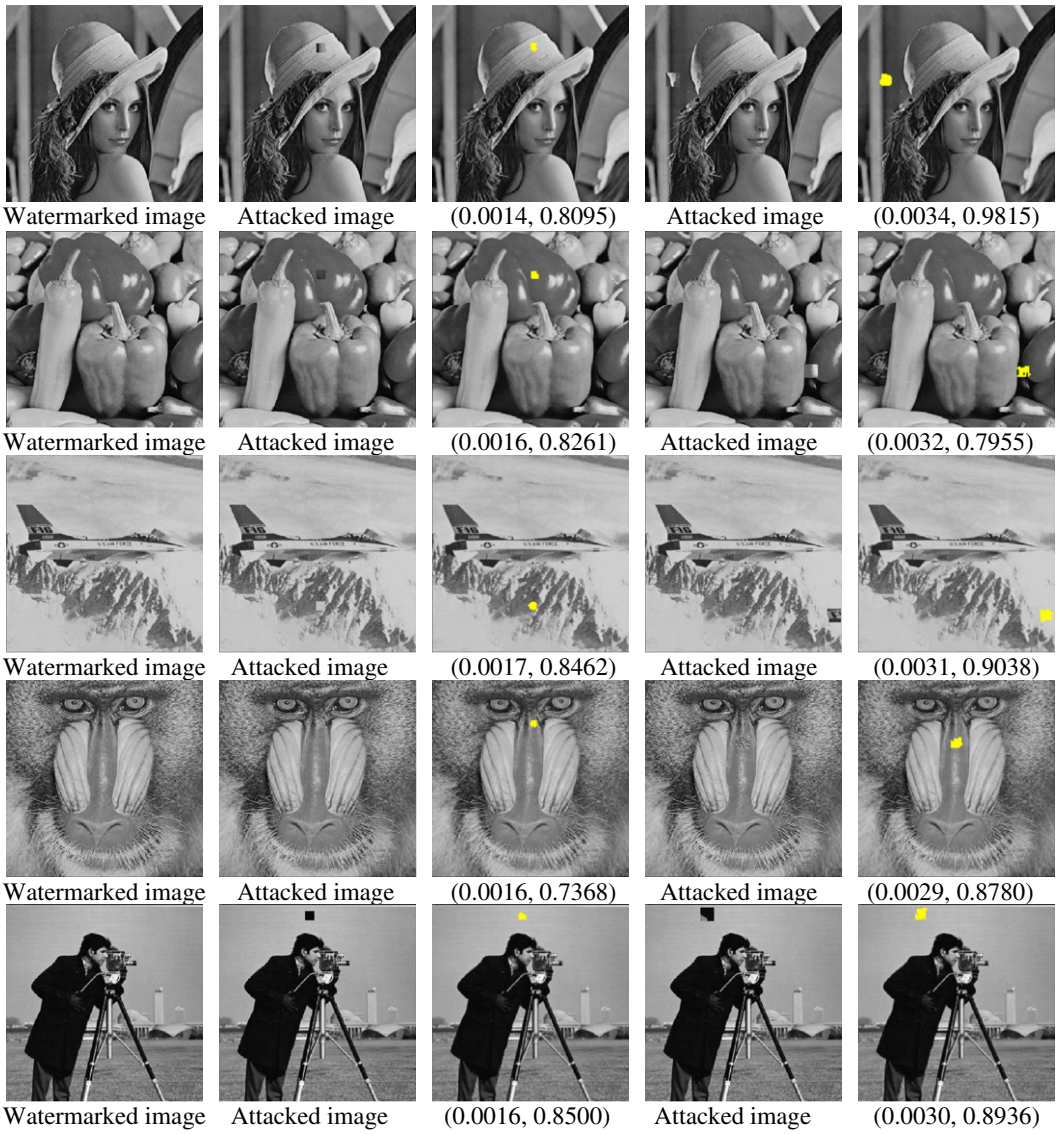

Fig. 14. Illustration of our tamper localization results: watermarked images (1st column), maliciously attacked images by substituting a 24×24 block (2nd column), our corresponding detected distortion regions (3rd column), maliciously attacked images by substituting a 36×36 block (4th column), and our corresponding detected distortion regions (5th column). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(modifying) an object using Photoshop. These maliciously attacked images were then saved as JPG images using the default compression setting. We also list the M_1 and M_2 values in a pair below each localization result. This figure clearly shows that our scheme successfully localizes the tampered regions. Based on our predefined thresholds, we conclude that our scheme detects all these maliciously attacked watermarked images as maliciously tampered.

Fig. 14 demonstrates the tampering localization results on five representative watermarked images, which were maliciously attacked by performing the 24×24 block or 36×36 block substitution followed by 80% JPEG compression. We do not list the localization results for 48×48 block substitution since their tampering localization results are always better than the small block substitution attacks. In Fig. 14, we also list the M_1 and M_2 values in a pair below each localization result. It clearly shows that our scheme successfully localizes the tampered regions when our authentication results indicate the forgery attacks.

4.5. Detection statistics under different simulated attacks

Finally, we evaluated the performance of our proposed scheme, the variant of our proposed scheme in the spatial domain, and four peer systems under two *PSNR* values (e.g., 38 db and 41 db) for all simulated attacks (e.g., image blurring, Gaussian low-pass filtering, median filtering, salt and peppers noise, JPEG compression, JPEG2000 compression, and substitution attacks) and each specific category of simulated attacks, respectively. For the substitution attacks, we performed each of the three kinds of block substitutions (e.g., 24×24, 36×36, and 48×48 block substitutions) followed by 80% JPEG compression 50 times on each of the 200 watermarked images. Tables 2 and 3 compare the performance of six semi-fragile watermarking algorithms in terms of true positive, false positive, false negative, and true negative under all simulated attacks for 200 watermarked images with *PSNR* values of around 41 db and 38 db, respectively. The two tables clearly show that our proposed scheme achieves the best performance with a high true positive of 90.6% and 93.5%, a high true negative of 65.3% and 70.7%, a low false negative of 9.4% and 6.5%, and a low false positive of 34.7% and 29.3% under *PSNR* values of 41db and 38db, respectively. The other schemes achieve worse performance mainly due to their less robustness to JPEG compression. Our proposed scheme achieves better performance than our variant mainly due to the choice of the wavelet transform domain as the embedding media. As the *PSNR* values decrease from 41 db to 38 db, most schemes tend to perform better due to the increasing robustness to common image processing attacks.

Tables 4 and 5 compare the performance of six semi-fragile watermarking algorithms in terms of their miss probability under each of the three kinds of malicious attacks and in terms of false alarm probability under each of common image processing attacks that preserves the content of the image for 200 watermarked images with *PSNR* values of 41 db and 38 db, respectively. The two tables clearly show that our proposed scheme achieves the smallest probability of false alarm for each incidental attack and the smallest probability of miss for each substitution attack. Specifically, our scheme is 100% robust against blurring, Gaussian low-pass filtering, and salt and peppers noise attacks and is more fragile to relatively large malicious attacks. The miss probability increases when the substitution block size decreases. This is reasonable due to small changes resulted from the small substitution block size.

Table 2
Detection results of six semi-fragile watermarking schemes under all simulated attacks for 200 watermarked images with the *PSNR* value of around 41 db.

Methods	Actual incidental attacks						Actual malicious attacks					
	Ours (%)	Variant (%)	Yang (%)	Che (%)	Maeno (%)	Cruz (%)	Ours (%)	Variant (%)	Yang (%)	Che (%)	Maeno (%)	Cruz (%)
Detected incidental attacks	90.6	83.3	38.4	27.6	16.7	55.6	34.7	100	65.8	59.3	100	100
Detected malicious attacks	9.4	16.7	61.6	72.4	83.3	44.4	65.3	0	34.2	40.7	0	0

Table 3
Detection results of six semi-fragile watermarking schemes under all simulated attacks for 200 watermarked images with the *PSNR* value of around 38 db.

Methods	Actual incidental attacks						Actual malicious attacks					
	Ours (%)	Variant (%)	Yang (%)	Che (%)	Maeno (%)	Cruz (%)	Ours (%)	Variant (%)	Yang (%)	Che (%)	Maeno (%)	Cruz (%)
Detected incidental attacks	93.5	84.7	39.6	41.5	30.9	63.3	29.3	100	48.1	33.3	49.6	100
Detected malicious attacks	6.5	15.3	60.4	58.5	69.1	36.7	70.7	0	51.9	66.7	50.4	0

Table 4
Detection results of six semi-fragile watermarking schemes under each simulated attack for 200 watermarked images with the *PSNR* value of around 41 db.

Method	Attack								
	Probability of false alarm						Probability of miss		
	Blur (%)	Gaussian (%)	Median (%)	S&P (%)	JPEG (%)	JPEG2k (%)	24×24 (%)	36×36 (%)	48×48 (%)
Ours	0	0	39.6	0	9.8	2.3	59.2	31.6	13.8
Spatial	8.6	0	73.5	0	14.7	2.7	100	100	100
Yang	100	76.4	94.7	8.4	54.3	9.2	71.2	66.8	59.6
Che	100	64.3	94.5	88.4	66.6	30.2	100	40.9	40.3
Maeno	100	100	100	100	90.5	22.4	100	100	100
Cruz	88.9	8.5	88.6	40.3	24.3	16.9	100	100	100

Table 5
Detection results of six semi-fragile watermarking schemes under each simulated attack for 200 watermarked images with the *PSNR* value of around 38 db.

Method	Attack								
	Probability of false alarm						Probability of miss		
	Blur (%)	Gaussian (%)	Median (%)	S&P (%)	JPEG (%)	JPEG2k (%)	24 × 24 (%)	36 × 36 (%)	48 × 48 (%)
Ours	0	0	34.3	0	0	1.7	51.6	26.3	10.4
Spatial	4.3	0	69.8	0	2.4	1.9	100	100	100
Yang	100	68.6	93.5	0	50.9	8.4	59.6	48.4	36.8
Che	96	40.1	92.3	60.7	46.2	18.8	40.3	39.1	20.4
Maeno	100	84.6	100	40.4	70.1	2.9	100	50.4	20.8
Cruz	82.4	2.9	78.2	38.9	12.8	8.4	100	100	100

None of the four peer schemes achieves the comparable performance as our scheme since they usually cannot distinguish these malicious attacks from the JPEG compression attacks (i.e., they usually classify the watermarked images under these substitution attacks as either authentic or incidental distorted or non-presence of watermark). Our variant scheme achieves the second best robustness against incidental attacks. However, it completely fails to detect malicious attacks smaller than a block of 48 × 48 mainly due to the low clustering level of tampered error pixels, which result from the choice of the spatial domain as the embedding media. Both tables clearly show that most schemes tend to perform better when the *PSNR* values decrease from 41db to 38db since they are more robust against common image processing attacks.

5. Conclusions

We present a novel semi-fragile watermarking scheme for image content authentication with tampering localization. The contributions of the proposed scheme are:

- Applying the quantization method to embed the private-key-based watermark in the wavelet domain so that a majority of image distortions, which cause the intensity shift by a value larger than a half of the quantizer q, can be detected in the authentication process. Unlike traditional quantization based approaches, our quantization method modifies only one chosen wavelet coefficient in the approximation subband of the Haar wavelet transform of each block (i.e., embeds a watermark bit in one chosen approximation coefficient) to ensure its robustness against moderate incidental attacks and fragileness against malicious attacks. In addition, our quantization method extracts the watermark bit using the round operation instead of the traditional floor operation to further ensure the semi-fragile property.
- Defining two kinds of tampered error pixels (e.g., strongly tampered and mildly tampered error pixels) and two authentication measures to detect the authenticity of the probe image and prove tampering. Specifically, we consider an error pixel as strongly tampered if at least four of its eight neighbors are error pixels and an error pixel as mildly tampered if less than four of its eight neighbors are error pixels. We compute the percentage of error pixels in the error map as the first authentication measure, M_1, which quantitatively evaluates the overall similarity between extracted and embedded watermarks. We compute the ratio of the number of our defined strongly tampered error pixels to the number of our defined tampered error pixels in the error map as the second authentication measure, M_2, which evaluates the overall clustering level of the tampered error pixels.
- Using a binary error map together with the two authentication measures in the authentication process to compensate the possible misclassification in the error map, capture all possible distortions, and localize all possible tampered areas.
- Applying randomness strategies to increase the security of the proposed system. To this end, we first apply the Mersenne Twister algorithm to generate a watermark bit sequence using a private key. We then apply the one-way hash function to choose the order of the blocks for embedding watermark using two secret keys. We finally apply the Mersenne Twister algorithm to generate the embedding positions for each block using three private keys.

Our extensive experimental results show that the proposed scheme successfully distinguishes malicious attacks from non-malicious tampering of image content. It also accurately localizes maliciously tampered regions. Our scheme is more robust to acceptable content preserving operations and more fragile to malicious distortions than four semi-fragile watermarking schemes and its variant.

Our future work includes studying the tampering detection sensibility when an image size changes, addressing geometric attack issues, and testing more images of various types.

Appendix A. The Φ function in Eqs. (7) and (8) follows a standard normal distribution

We derive the threshold for incidental attacks in Eq. (7) using Gaussian distribution. As explained in Section 2.3, we know that the probability for a pixel in the error map to be detected as 0 or 1 is 0.5. So, each pixel is a binomially distributed random variable. The probability is the same as the probability that a tossed coin will come up heads. Let N_{total} represent the number of times the coin tossed and N represent the number of times that the coin comes up heads. The probability that heads will appear N_h times is:

$$P(N = N_h) = \binom{N_{total}}{N_h} p^{N_h}(1-p)^{N_{total}-N_h}$$
$$\text{where } \binom{n}{m} = \frac{n!}{m!(n-m)!}, \quad 0 \leqslant m \leqslant n,$$
$$p = 0.5 \tag{13}$$

That is, Eq. (13) can be directly used to calculate $P(N = N_h)$. Fig. 15(a) shows the distributions for Eq. (13) when $N_{total} = 100$. As a result, we conclude that a binomial probability variable can be approximated using a Gaussian probability variable and the error distribution follows a Gaussian distribution. Furthermore, this Gaussian distribution, represented by Φ function in Eq. (7), approaches the normal distribution with expected value 0 and variance 1 when N_{total} is large (i.e., the total number of pixels in *ErrorMap* is large). So, we claim that Φ function used in Eq. (7) follows a standard normal distribution.

Similarly, we derive the threshold for malicious attacks in Eq. (8) using Gaussian distribution. In our system, each pixel is a binomially distributed random variable. As a result, the probability for

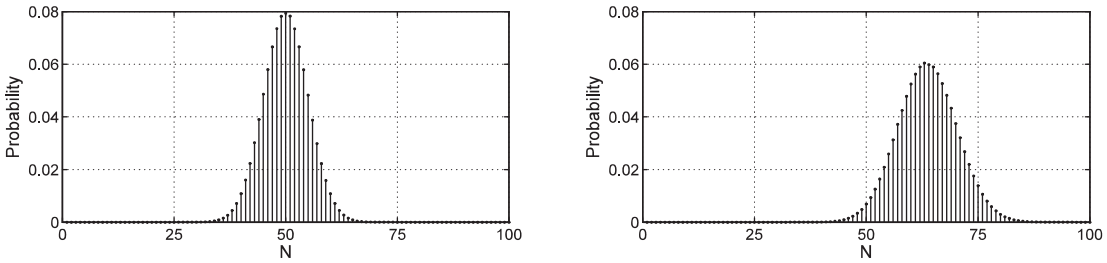

Fig. 15. The probability density for two cases. (a) The number of pixels in the error map (i.e., the number of times the coin tossed) is 100. (b) The number of possible tampered error pixels is 100.

a pixel in the error map to be detected as tampered error pixels is 0.4980 and the probability for a pixel in the error map to be detected as a strongly tampered pixel is 0.3184. When considering N_{total} possible tampered error pixels, the probability that a strongly tampered error pixel will appear N_h times is:

$$P(N = N_h) = \binom{N_{total}}{N_h} p^{N_h}(1-p)^{N_{total}-N_h}$$
$$\text{where } \binom{n}{m} = \frac{n!}{m!(n-m)!}, \quad 0 \leqslant m \leqslant n, \quad p = 0.3184/0.4880 = 0.6394 \tag{14}$$

Fig. 15(b) shows the distributions for Eq. (14) when N_{total} = 100. As a result, we conclude that the ratio of strongly tampered error pixels to tampered error pixels can be approximated using a Gaussian distribution. Furthermore, this Gaussian distribution, represented by Φ function in Eq. (8), approaches the normal distribution with expected value 0 and variance 1 when N_{total} is large (i.e., the total number of tampered error pixels in *ErrorMap* is large). So, we claim Φ function used in Eq. (8) follows a standard normal distribution.

References

[1] E. Lin, C. Podilchun, E. Delp, Detection of image alterations using semi-fragile watermarks, in: Proceedings of SPIE International Conference on Security and Watermarking of Multimedia Contents, 2000, pp. 152–163.

[2] C. Lin, S. Chang, Semi-fragile watermarking for authenticating JPEG visual content, in: Proceedings of SPIE on Security and Watermarking of Multimedia Content, 2000, pp. 140–152.

[3] C. Lin, S. Chang, A robust image authentication method distinguishing JPEG compression from malicious manipulation, IEEE Trans. Circuits Syst. Video Technol. 11 (2) (2001) 153–168.

[4] C. Ho, C. Li, Semi-fragile watermarking scheme for authentication of JPEG images, in: Proceedings of International Conference on ITCC, 2004, pp. 7–11.

[5] K. Maeno, Q. Sun, S. Chang, M. Suto, New semi-fragile image authentication watermarking techniques using random bias and nonuniform quantization, IEEE Trans. Multimedia 8 (1) (2006) 32–45.

[6] J. Eggers, B. Girod, Blind watermarking applied to image authentication, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001, pp. 1977–1980.

[7] J. Fridrich, A hybrid watermark for tamper detection in digital images, in: Proceedings of the Fifth International Symposium on Signal Processing and Its Applications, 1999, pp. 301–304.

[8] T. Lan, M. Mansour, H. Liao, Robust high capacity data embedding, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001, pp. 581–584.

[9] D. Kundur, D. Hatzinakos, Digital watermarking for telltale tamper proofing and authentication, in: Proceedings of the IEEE: Special Issue on Identification and Protection of Multimedia Information, 1999, pp. 1167–1180.

[10] G. Yu, C. Lu, H. Liao, Mean quantization-based fragile watermarking for image authentication, Opt. Eng. 40 (7) (2001) 1396–1408.

[11] X. Zhou, X. Duan, D. Wang, A semi-fragile watermarking scheme for image authentication, in: Proceedings of the 10th International Conference on Multimedia Modeling, 2004, pp. 374–377.

[12] H. Kang, J. Park, A semi-fragile watermarking using JND, in: Proceedings of STEG, 2003, pp. 127–131.

[13] Y. Hu, D. Han, Using two semi-fragile watermarks for image authentication, in: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 2005, pp. 5484–5489.

[14] H. Liu, J. Lin, J. Huang, Image authentication using content based watermark, in: Proceedings of IEEE International Symposium on Circuits and Systems, 2005, pp. 4014–4017.

[15] Y. Zhu, C. Li, H. Zhao, Structural digital signature and semi-fragile fingerprinting for image authentication in wavelet domain, in: Proceedings of IAS, 2007, pp. 478-483.

[16] H. Yang, X. Sun, Semi-fragile watermarking for image authentication and tamper detection using HVS model, in: Proceedings of International Conference on Multimedia and Ubiquitous Engineering, 2007, pp. 1112–1117.

[17] S. Che, B. Ma, Z. Che, Semi-fragile image watermarking algorithm based on visual features, in: Proceedings of International Conference on Wavelet Analysis and Pattern Recognition, 2007, pp. 382–387.

[18] C. Cruz, R. Reyes, M. Nakano, H. Perez, Image content authentication system based on semi-fragile watermarking, in: Proceedings of the 51st Midwest Symposium on Circuits and Systems, 2008, pp. 306–309.

[19] T. Liu, Z. Qiu, The survey of digital watermarking-based image authentication techniques, in: Proceedings of the Sixth International Conference on Signal Processing, 2002, pp. 1556–1559.

[20] M. Matsumoto, T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator, ACM Trans. Model. Comput. Simul. 8 (1) (1998) 3–30.

[21] X. Qi, J. Qi, A robust content-based digital image watermarking scheme, Signal Process 87 (6) (2007) 1264–1280.

[22] A. Uhl, A. Pommer, Image and Video Encryption: From Digital Rights Management to Secured Personal Communication (Advances in Information Security), Springer, 2004. p. 103.

[23] M. Hsieh, D. Tseng, Perceptual digital watermarking for image authentication in electronic commerce, Electron. Commer. Res. (4) (2004) 157–170.

[24] O. Ekica, B. Sankur, B. Coskun, U. Naci, M. Akcay, Comparative evaluation of semifragile watermarking algorithms, J. Electron. Imag. 13 (1) (2004) 209–216.